AU 332 Artificial Intelligence: Principles and Techniques

By: Shi Rui (ID: 518030910397)

HW#: 4

November 26, 2020

I. CODING PART

A. joinFactors

Function joinFactors(factor1, factor2) return a factor table that is the join of factor1 and factor2. We add a column 'common' to both factor1 and factor2 in case they do not share a same column. Then we utilize pandas.merge to combine two factors, and drop redundant column 'probs_y' and 'common'.

```
def joinFactors(factor1, factor2):

    f1 = pd.DataFrame.copy(factor1)
    f2 = pd.DataFrame.copy(factor2)
    f1['commom'] = 1
    f2['commom'] = 1

    intersection = list((f1.columns).intersection(f2.columns))
    intersection.remove('probs')

    joinFactor = pd.merge(f1, f2, on=intersection, how='outer')
    joinFactor['probs_x'] *= joinFactor['probs_y']
    joinFactor = joinFactor.drop(columns=['probs_y','commom'], axis=1)
    joinFactor = joinFactor.rename(columns={'probs_x' : 'probs'})

    return joinFactor
```

B. marginalizeFactor

Function marginalizeFactor(factorTable, hiddenVar) returns a factor table that marginalizes marginal variable hiddenVar out of factorTable. If the hiddenVar is not in the columns of factorTable, return the original factorTable directly. Then drop the column of hiddenVar. If the factortable only has columns other than prob, utilize pandas.groupby to group up the factorTable, otherwise, return the factor directly.

```
def marginalizeFactor(factorTable, hiddenVar):

    factor = pd.DataFrame.copy(factorTable)

    if hiddenVar not in list(factor.columns):
        return factor

    factor = factor.drop(columns=hiddenVar, axis=1)
    var = list(factor.columns)
    var.remove('probs')

    if not var:
        return factor
    else:
        factor = factor.groupby(var, as_index=False).sum()

    return factor
```

C. marginalizeNetworkVariables

Function marginalizeNetworkVariables(bayesNet, hiddenVar) takes a Bayesian network, bayesNet, and marginalizes out a list of variables hiddenVar. First, we check whether hiddenVar is a string of variable or a list of strings of variables. Then for each hidden variable var we intend to marginalize, we join all the factors with a column of var in the bayesNet, and use Function marginalizeFactor to marginalize var out of the

joined factor. The marginalized factor and factors without column of var are combined as a new bayesNet for the next variable.

```
def marginalizeNetworkVariables(bayesNet, hiddenVar):

    if isinstance(hiddenVar, str):
        hiddenVar = [hiddenVar]

    if not bayesNet or not hiddenVar:
        return bayesNet

    marginalizeBayesNet = bayesNet.copy()

    for var in hiddenVar:
        tmp = []
        tmpfactor = None
        for factor in marginalizeBayesNet:
            if var in factor.columns:
                tmpfactor = factor if tmpfactor is None else joinFactors(factor, tmpfactor)
            else:
                tmp.append(factor)
        if tmpfactor is not None:
            tmp.append(marginalizeFactor(tmpfactor, var))
        marginalizeBayesNet = tmp.copy()

    return marginalizeBayesNet
```

D. evidenceUpdateNet

Function evidenceUpdateNet(bayesNet, evidenceVars, evidenceVals) takes a Bayesian network bayesNet and sets the list of variables evidenceVars to the corresponding list of values evidenceVals. In other words, it sets the value of evidence variables and remove other values. And normalization is not required. For each evidence variable var and its corresponding value val, for each factor in bayesNet with a column of var, save the row where the value of var is equal to val and drop other rows.

```
def evidenceUpdateNet(bayesNet, evidenceVars, evidenceVals):

    if isinstance(evidenceVars, str):
        evidenceVars = [evidenceVars]
    if isinstance(evidenceVals, str):
        evidenceVals = [evidenceVals]

    updatedBayesNet = bayesNet.copy()
    for idx in range(len(evidenceVars)):
        variable = evidenceVars[idx]
        value = int(evidenceVals[idx])
        tmpnet = updatedBayesNet.copy()
        updatedBayesNet = []

        for factorTable in tmpnet:
            if variable in factorTable.columns:
                factorTable = factorTable[factorTable[variable]==value]
            updatedBayesNet.append(factorTable)

    return updatedBayesNet
```

E. inference

Function inference(bayesNet, hiddenVar, evidenceVars, evidenceVals) takes in a Bayesian network and returns a single joint probability table resulting from the given set of evidence variables and marginalizing a set of hidden variables. Normalization to give valid probabilities is required, and The final table should be a proper probability table.

```
def inference(bayesNet, hiddenVar, evidenceVars, evidenceVals):

    if not bayesNet:
        return bayesNet

    inferenceNet = bayesNet.copy()

    inferenceNet = evidenceUpdateNet(inferenceNet, evidenceVars, evidenceVals)
    inferenceNet = marginalizeNetworkVariables(inferenceNet, hiddenVar=hiddenVar)

    length = len(inferenceNet)
    if length == 1:
        factor = inferenceNet[0]
    else:
        factor = inferenceNet[0]
        for idx in range(1, length):
            factor = joinFactors(factor, inferenceNet[idx])

    # normalization
    norm = sum(list(factor['probs']))
    factor['probs'] /= norm

    return factor
```

II. WRITTEN PART

A. Question 1

$8+8\times2+8\times2+8\times2+8\times2+8\times2\times2\times4+8\times2\times2\times2\times2\times4+8\times2\times2\times2\times2+4\times4+4\times4\times2\times2+4\times4\times2\times2+4\times4\times2\times2=1048$

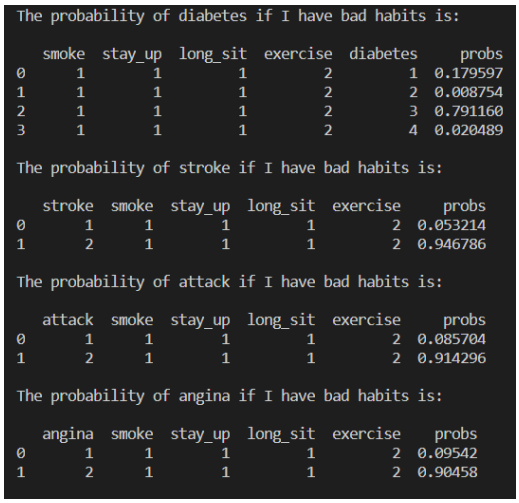
Thus, the size of network is 1048.

B. Question 2

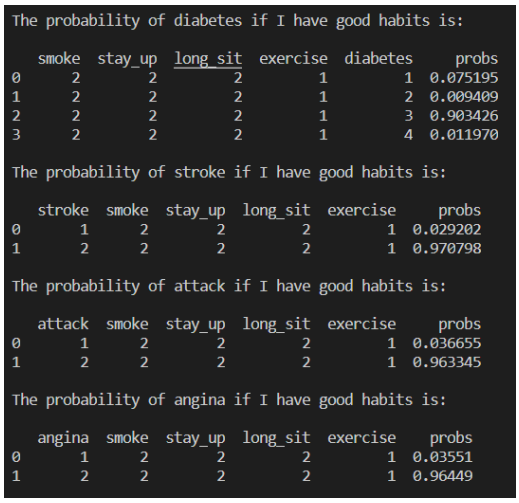
The answer is shown in TABLE I. And the output of code is shown in FIG.1.

health outcomes		bad habits	good habits	pool health	good health
diabetes	1	0.179597	0.075195	0.115423	0.057710
	2	0.008754	0.009409	0.007662	0.009543
	3	0.791160	0.903426	0.860873	0.922194
	4	0.020489	0.011970	0.016043	0.010553
stroke	1	0.053214	0.029202	0.082686	0.01446
	2	0.946786	0.970798	0.917314	0.98554
heart attack	1	0.085704	0.036655	0.140784	0.016161
	2	0.914296	0.963345	0.859216	0.983839
angina	1	0.09542	0.03551	0.161608	0.013326
	2	0.90458	0.96449	0.838392	0.986674

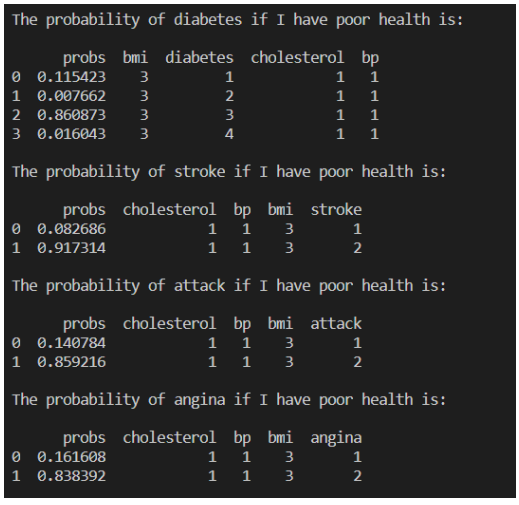
TABLE I: Answer of Question2



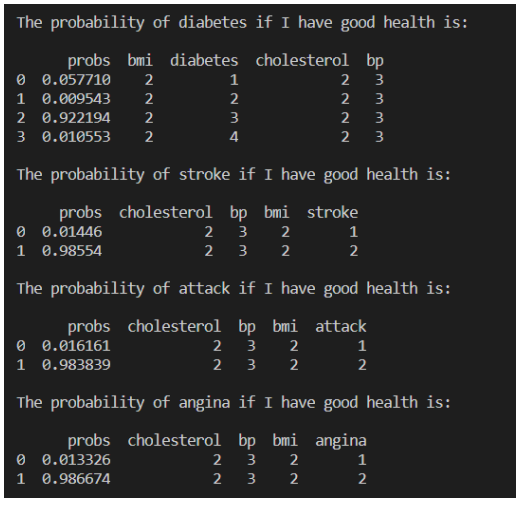
bad habit



good habit



poor health



good health

FIG. 1: Coding Result of Question2

III. QUESTION 3

The figure of probability of four health outcomes given income status is shown in FIG.2.

From the figures, it seems that people with higher income tends to have a lower probability to suffer health problems. With the income increases, the probabilities of diabetes, stroke, heart attack and angina decrease. However, the probability of stroke, heart attack and angina at income level 2 is highest, that is, people with income $\$10,000 - \$15,000$ is more likely to suffer from the diseases than people earn less than $\$10,000$.

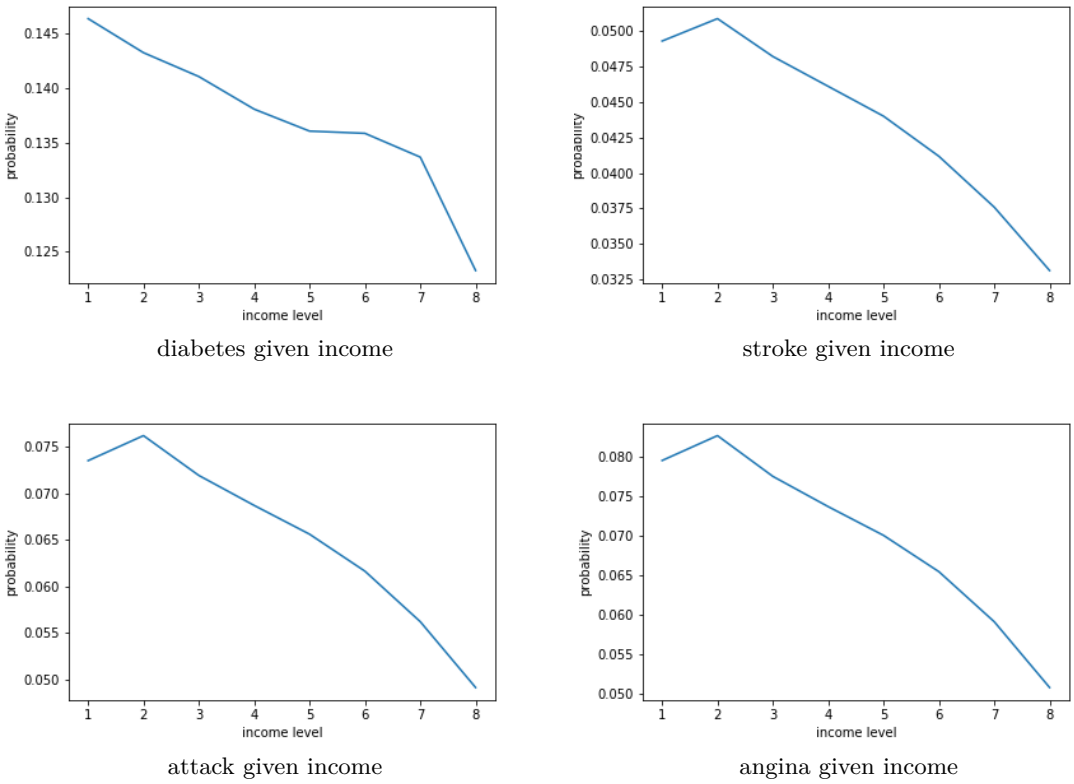


FIG. 2: Answer of Question3

IV. QUESTION 4

There are no links in the graph between habits and outcomes. Thus the assumption is that smoking and exercise has no direct effect on health problems. To test the validity of the assumption, we create a new Bayesian Network by adding edges from smoke to each of the four outcomes and edges from exercise to each of the four outcomes. The probability of health outcomes given bad/good habits and poor/good health on new network is shown in TABLE II. And the output of code is shown in FIG.3.

Compare TABLE II with TABLE I, we find there is significant difference on probability of health outcomes given habits, while the difference on probability of health outcomes given health is tiny. After adding edges from smoke to each of the four outcomes and edges from exercise to each of the four outcomes, the probability of health problem given bad habits is increased, while the probability of health problem given good habits is decreased. However, the probability of health problem given health change little. There are some dependence between habits and health outcomes, thus the assumption is not valid. But given health, the assumption that habits and health outcomes are independent is valid.

health outcomes		bad habits	good habits	pool health	good health
diabetes	1	0.245992	0.056227	0.121241	0.055937
	2	0.006928	0.010160	0.007492	0.009697
	3	0.723721	0.923710	0.854769	0.924042
	4	0.023359	0.009903	0.016498	0.010323
stroke	1	0.080488	0.019464	0.082689	0.014544
	2	0.919512	0.980536	0.917303	0.985456
heart attack	1	0.135301	0.021213	0.140083	0.016183
	2	0.864699	0.978787	0.859917	0.983817
angina	1	0.138072	0.023948	0.161096	0.013328
	2	0.861928	0.976052	0.838904	0.986672

TABLE II: Answer of Question4

```
The probability of diabetes if I have bad habits is:
   exercise  smoke  stay_up  diabetes  long_sit     probs
0         2      1        1         1         1  0.245992
1         2      1        1         2         1  0.006928
2         2      1        1         3         1  0.723721
3         2      1        1         4         1  0.023359

The probability of stroke if I have bad habits is:
   exercise  smoke  stroke  stay_up  long_sit     probs
0         2      1       1        1         1  0.080488
1         2      1       2        1         1  0.919512

The probability of attack if I have bad habits is:
   exercise  smoke  attack  stay_up  long_sit     probs
0         2      1       1        1         1  0.135301
1         2      1       2        1         1  0.864699

The probability of angina if I have bad habits is:
   exercise  smoke  angina  stay_up  long_sit     probs
0         2      1       1        1         1  0.138072
1         2      1       2        1         1  0.861928
```

bad habit

```
The probability of diabetes if I have good habits is:
   exercise  smoke  stay_up  diabetes  long_sit     probs
0         1      2        2         1         2  0.056227
1         1      2        2         2         2  0.010160
2         1      2        2         3         2  0.923710
3         1      2        2         4         2  0.009903

The probability of stroke if I have good habits is:
   exercise  smoke  stroke  stay_up  long_sit     probs
0         1      2       1        2         2  0.019464
1         1      2       2        2         2  0.980536

The probability of attack if I have good habits is:
   exercise  smoke  attack  stay_up  long_sit     probs
0         1      2       1        2         2  0.021213
1         1      2       2        2         2  0.978787

The probability of angina if I have good habits is:
   exercise  smoke  angina  stay_up  long_sit     probs
0         1      2       1        2         2  0.023948
1         1      2       2        2         2  0.976052
```

good habit

```
The probability of diabetes if I have poor health is:
   cholesterol  bp  bmi  diabetes     probs
0            1   1    3         1  0.121241
1            1   1    3         2  0.007492
2            1   1    3         3  0.854769
3            1   1    3         4  0.016498

The probability of stroke if I have poor health is:
   cholesterol  bp  bmi  stroke     probs
0            1   1    3       1  0.082697
1            1   1    3       2  0.917303

The probability of attack if I have poor health is:
   cholesterol  bp  bmi  attack     probs
0            1   1    3       1  0.140083
1            1   1    3       2  0.859917

The probability of angina if I have poor health is:
   cholesterol  bp  bmi  angina     probs
0            1   1    3       1  0.161096
1            1   1    3       2  0.838904
```

poor health

```
The probability of diabetes if I have good health is:
   cholesterol  bp  bmi  diabetes     probs
0            2   3    2         1  0.055937
1            2   3    2         2  0.009697
2            2   3    2         3  0.924042
3            2   3    2         4  0.010323

The probability of stroke if I have good health is:
   cholesterol  bp  bmi  stroke     probs
0            2   3    2       1  0.014544
1            2   3    2       2  0.985456

The probability of attack if I have good health is:
   cholesterol  bp  bmi  attack     probs
0            2   3    2       1  0.016183
1            2   3    2       2  0.983817

The probability of angina if I have good health is:
   cholesterol  bp  bmi  angina     probs
0            2   3    2       1  0.013328
1            2   3    2       2  0.986672
```

good health

FIG. 3: Coding Result of Question4

V. QUESTION 5

There is no edge between four outcomes. Thus the assumption is that one outcome has no effects on the other outcomes. To test the validity of the assumption, we create a new Bayesian Network by adding an edge from diabetes to stroke. The result is shown in FIG.4. In the second network,

$$P(stroke = 1|diabetes = 1) = 0.044417$$
$$P(stroke = 1|diabetes = 3) = 0.039955$$

In the third network, adding an edge from diabetes to stroke

$$P(stroke = 1|diabetes = 1) = 0.076542$$
$$P(stroke = 1|diabetes = 3) = 0.034456$$

```
Question5 ------------------------------------------------------------
second network:
probability of stroke level 1 given diabetes level 1 is 0.044417
probability of stroke level 1 given diabetes level 3 is 0.039955
third network: Adding an edge from diabetes to stroke
probability of stroke level 1 given diabetes level 1 is 0.076542
probability of stroke level 1 given diabetes level 3 is 0.034456
```

FIG. 4: Answer of Question5

The result shows that a person suffering from diabetes is more likely to suffer from stroke after adding an edge from diabetes to stroke. And a person without diabetes is less likely to suffer from stroke after adding an edge from diabetes to stroke. Thus diabetes has some effect on stroke, the assumption is invalid.